2019(107-2) 計概期中Project

一、題目:

Concordance

二、程式說明:

“請一定要使用老大規定的資料結構”(下一篇 i-learning上都有)

這個project是要將讀入的程式碼切成一個一個的token，

且將每個token根據下列3個case做分類。

Case1：Identifier

底線 或 英文字母 開頭，之後跟著英文字母或底線或數字

Ex： int, \_abc, k9999, a, cc, kfn\_0, a2b, return, ...

Case2：數字 與 字串

數字可能會有小數點，字串(被 2 個 " 夾起來(不跨行))

Ex： 1, 565, 5.22, 0.2, "hello 5.22", "982\_final&& , "code: \"\"" ...

字元 (被 2 個 ' 夾起來(不跨行))

Ex : 'a' 'b' 'c' '\n' '\'' -> 不會有 compile 不會過的數據

Case3：特殊符號 ( 以下所列出來的38種特殊符號 )

+ - \* / > < = <= >= == != ? % & &&

|| | ^ " ' . , ( ) [ ] { } << >>

! : ; # -> ++ -- +=

＊ 該如何切 token ?

從下一個( 尚未讀進來的 或 「已讀進來但尚未處理」的 )字元開始

// 反正就是在概念上尚未處理到它的那個字元

// 小心：輸入的程式碼最後會以'END\_OF\_FILE'這個字做結尾

1. 跳過所有white-space及註解、直到碰到「非white-space」

(或'END\_OF\_FILE'這個字)為止

// 什麼是white-space ? 空白,Tab,換行

// 什麼是註解 ? 「//」之後直到end line 或 夾在「/\*」與「\*/」之間的

2. 從那個「非white-space」開始讀入、且盡可能的往後讀，

直到碰到「white-space」 或 「前面所提到的特殊字元」 或 「註解」為止

// 注意！後來碰到的那個「前面所提到的特殊字元」 就是之前所謂

「已讀進來但尚未處理」的字元！

3. 此時你已讀入一個token了，問題只是「它是哪個case」而已

Ex：

Input:

void main ( )

{

int a,b ;

a = 10 ;

b = 20 ;

b = a+b ;

} // end main( )

END\_OF\_FILE

依據Case做分類，結果如下：

Identifier :

void main int a b a b b a b

數字 :

10 20

特殊符號 :

( ) { , ; = ; = ; = + ; }

◎ 接著要完成以下功能：

1.總共有多少個 token

2.三種case各有多少 token

3.印出指定 token 的位置 (line number,column number)

4.印出所指定的那一 line 出現哪幾個 token (要排序)

5.結束

三、Input格式說明:

Input的方式採用螢幕輸入。

會輸入一段長度不定的程式碼，以 'END\_OF\_FILE' 這個字做結尾。

接著根據上述功能會輸入數個1~5的數字及要求的input，來測試互動式功能選項。

Input格式範例:

void main ( )

{

... // 程式碼

}

END\_OF\_FILE // 最後會以 'END\_OF\_FILE' 這個字做結尾

1 // '1'表示請輸出總共有多少個token

3 main // '3 main'表示印出"main"這個token出現的位置

5 // '5'表示結束

四、Output格式說明:

參照 { 計概板8466 } 的方法做

你的輸出會產生在 { output.txt } 這檔案中

預設點擊兩下，會使用 { 記事本 } 檢視 { output.txt }

你會發現程式碼輸出 { \n } 的並不會有換行的效果

要使用 { NotePad++ } 做檢視動作，避免沒有換行效果

輸出中文出現亂碼的時候：可以在 NotePad++ -> 編碼 -> UTF-8

{ NotePad++ } 下載網址: http://www.azofreeware.com/2006/10/notepad-35.html

程式一開始必須把所有的 Token 都輸出出來

輸出的資料以座標形式表達 (X,Y) // 座標從1開始

X 座標為該 token 出現的 line number (行數)

Y 座標為該 token 字首在該行出現的 column (第幾個字元，含空白)

例如: void (1,11) // 表示 " void " 的所在位置Line為1, Column為11

// Token的名字跟後面的()必須要有一個空白

輸出順序依照 Token 的 ASCII 大小排列，由小到大，

如果第一個字母一樣就比較第2個字母，依此列推。

若同一個Token出現多次，則後面的座標形式先依照Line由小到大排列

若Line相同再依照Column由小到大排列

()跟()間不需要有空白

※ 注意!! 空白行也算一行，那行如果只有註解也算一行。

接著輸出 5 個互動式功能選項 // 這五個一定要輸出

1.總共有多少種 token

2.三種case各有多少 token

3.印出指定 token 的位置 (line number,column number)

4.印出所指定的那一 line 出現哪幾個 token (要排序)

5.結束

當使用者輸入 1 時，螢幕輸出 -> 總共 X 種

2 時，螢幕輸出 -> 各 Case 的 Token 數量

3 時，螢幕輸出 -> 要指定哪個 Token? -> 輸入要搜尋的 Token

-> 螢幕輸出 Token位置 (X1,Y1)(X2,Y2) ...

4 時，螢幕輸出 -> 要指定哪一Line? -> 輸入要指定的Line

-> 螢幕輸出那一Line出現的Token (要按ASCII排序)

5 時，螢幕輸出 -> byebye -> 結束

PS: byebye 後面不用換行!!!!!

只會出現 1~5 這些數字指令

以下只是為了讓你看懂，正確的格式請參照i-learning上提供的I/O檔案

範例 :

// ======= 以下為input ======= //

void main ()

{

int a,b ;

a = 10 ;

b = 20 ;

b = a+b ;

} // end main()

END\_OF\_FILE

// ======= 以上為input ======= //

// ======= 以下是在互動選單輸入功能 ======= //

1

2

3 main

4 5

5

// ======= 以上是在互動選單輸入功能 ======= //

// ======= 以下為螢幕所顯示的「程式輸出與使用者輸入」的範例 ======= //

( (1,11)

) (1,12)

+ (7,8)

, (4,8)

10 (5,7)

20 (6,7)

; (4,11)(5,10)(6,10)(7,11)

= (5,5)(6,5)(7,5)

a (4,7)(5,3)(7,7)

b (4,9)(6,3)(7,3)(7,9)

int (4,3)

main (1,6)

void (1,1)

{ (2,1)

} (9,1)

歡迎使用互動式功能選項 // 處理完一個指令之後，要再輸出一次這五行

1.總共有多少種 token

2.三種case各有多少 token

3.印出指定 token 的位置 (line number,column number) (要排序)

4.印出所指定的那一 line 出現哪幾個 token (要排序)

5.結束

> 1

總共15種

> 2

Case1 共 5個 // 這裡輸出的『共』

Case2 共 2個 // 之後空格加數字共要5個字元

Case3 共 8個 // 之後才接上『個』

> 3

請輸入要搜尋的 Token : main // 這行的 main 是使用者輸入

main (1,6) // 如果找不到k這個Token，輸出

> 4 // "查無此token: k"

請輸入要指定的 line : 5 // 這行的 5 是 使用者輸入

10

;

=

a

> 5

byebye//=== 以上為螢幕所顯示的「程式輸出與使用者輸入」的範例 === //

PS: 切記 byebye 後面沒有換行

五、作業繳交說明:

　　之後會另行公告